

# An object store for Fog infrastructures based on IPFS and a Scale-Out NAS

Bastien Confais  
CNRS, LS2N, UMR 6004,  
Polytech Nantes,  
rue Christian Pauc, BP 50609,  
44306 Nantes Cedex 3, France

Adrien Lebre  
INRIA, LS2N, UMR 6004,  
IMT-A,  
4 Rue Alfred Kastler,  
44300 Nantes, France

Benoît Parrein  
Université de Nantes, LS2N, UMR 6004,  
Polytech Nantes,  
rue Christian Pauc, BP 50609,  
44306 Nantes Cedex 3, France

**Abstract**—The Cloud Computing approach concentrates the computing power in few datacenters. The high latency to reach the platform makes this architecture not well suited for the Internet of Things. The Fog and Edge Computing propose to place servers near the users. In this context, we propose a first-class object store service for Fog/Edge facilities. Our proposal is built with Scale-out Network Attached Storage systems (NAS) and IPFS, a BitTorrent-based object store spread throughout the Fog/Edge infrastructure. Without impacting the IPFS advantages particularly in terms of data mobility, the use of a Scale-out NAS on each site reduces the inter-site exchanges that are costly but mandatory for the metadata management in the original IPFS implementation. Several experiments conducted on Grid’5000 testbed are analysed and confirmed, first, the benefit of using an object store service spread at the Edge and second, the importance of mitigating inter-site accesses. The paper concludes by giving a few directions to improve the performance and fault tolerance criteria of our Fog/Edge Object Store Service.

## I. INTRODUCTION

The Cloud Computing is not able to provide low latency computing that is necessary for the Internet of Things. The Fog Computing aims to deploy a lot of small datacenters geographically spread at the Edge of the network, close to the users to be reached with a quite low latency [1]. In this context, we are interested in the storage service that may be used in a such infrastructure. After having proposed a list of properties a storage system should have to be used in a Fog environment, we have shown that Interplanetary FileSystem (IPFS) was the best candidate to be used in a Fog environment [2]. Nevertheless, some improvements should be done such as reducing the amount of network traffic sent between the different sites of Fog.

In this work, we present our use of IPFS used on top of a Scale-Out NAS (Network Attached Storage) in Section II. Then, in Section III, we measure the performance of our approach on the Grid’5000 testbed not only by increasing the network latencies between the Fog sites but also by limiting the network throughput of the clients. Finally, Section IV concludes and gives some perspectives.

We limit our evaluation study to local access, when objects are stored on the closest site of the users.

## II. STORAGE SOLUTIONS

Interplanetary FileSystem (IPFS) [3] is an object store based on a Kademlia distributed hash table (DHT) used to store the

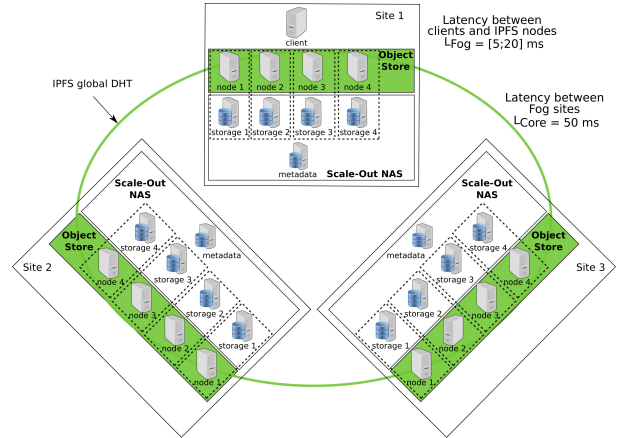


Fig. 1: Topology used to deploy an object store on top of a Scale-Out NAS local to each site.

location of the objects and protocol similar to BitTorrent to transfer the data between the nodes.

The problem of IPFS when it is deployed in a Fog Computing environment is that the global DHT containing the location of every object does not provide any locality. When an object is requested from a node which does not store it, the node has to locate the object by sending a request outside the Fog site, even if the object is finally stored on another node located on the same site. The request sent outside the site is a major problem because it increases the latencies and avoids the users to access the objects stored on the site in case of network partitioning.

To solve this problem, we proposed an original idea consisting in deploying a distributed filesystem locally on each site. This filesystem is used by IPFS to store objects. The interest of this coupling is that objects stored in a site are available directly to all the IPFS nodes of the site. The DHT is only used to locate objects that are not stored on the local site.

Our solution avoids the interactions with the remote sites. A such deployment is illustrated in Figure 1, where all the clients are located on a same site. We show that the overhead related to the use of the distributed file system is quite low in writing. In reading, not to access the global DHT reduces significantly the access times.

	Workload	Mean <b>writing</b> time (seconds)			Mean <b>reading</b> time (seconds)		
		5ms	10ms	20ms	5ms	10ms	20ms
IPFS+ RozoFS	100 × 256KB	0.455	0.479	0.621	0.380	0.489	0.687
	100 × 1MB	1.541	1.585	1.782	1.401	1.493	1.894
	100 × 10MB	15.156	15.022	14.847	14.708	15.099	15.029
IPFS Scale	100 × 256KB	0.473	0.490	0.730	0.397	0.531	0.776
	100 × 1MB	1.647	1.644	1.855	1.403	1.515	1.934
	100 × 10MB	15.260	15.240	15.410	14.102	14.763	15.326

TABLE I: Mean time to read or write objects with a size of {256KB, 1MB and 10MB} in function of the  $L_{Fog}$  latency. Only the workloads using 100 objects are represented.

### III. EXPERIMENTATION

In this section, we compare the performance of IPFS used alone and IPFS coupled with the Scale-Out NAS RozoFS [4].

#### A. Material and Method

The evaluation of IPFS and IPFS coupled with RozoFS is performed on the Grid'5000 testbed. RozoFS is an open-source distributed filesystem providing good performance in both sequential and random access. The topology in Figure 1 is considered, composed of 3 sites. Each of them has 4 IPFS nodes which are also RozoFS nodes and a metadata server for RozoFS. All the clients (between 1 and 10) are connected to the same site of Fog and we focus only on the local access: objects are stored on the site the clients are connected to. The scenario consists for the clients to write all the objects on the site and to they read them. The IPFS node each request is sent to is selected randomly by the clients and all the requests are sent simultaneously. We measure the time to write and read the objects. We vary the number of clients and also the network latency between the clients and the IPFS nodes ( $L_{Fog}$ ) between 5 and 20 ms [5], [6]. The network latency between the Fog sites is set to  $L_{Core} = 50$  ms and the latency between the nodes of a same site is set to 0.5 ms. Latencies are emulated artificially thanks to the Linux Traffic Control Utility (tc). To get a more realistic scenario, we also limited the throughput of the clients to 512Mbps that is close to the throughput of a LTE network [6]. Metadata replication in IPFS is disabled and data are stored in a `tmpfs` to remove the potential biases from the underlying storage unit.

#### B. Results

Table I shows the access time for an object in function of the  $L_{Fog}$  latency when only one client writes and reads 100 objects on its site. The main observation is that the latency has a bigger impact on small objects (256KB) than big objects (10MB). It takes 0.380s to read an object of 256KB when  $L_{Fog} = 5$  ms and 0.687 s when  $L_{Fog} = 20$  ms whereas it takes 14.708 s and 14.102 s with 10MB objects. For the objects of 10MB, the impact of  $L_{Fog}$  is small compared to the time to transfer the object. We also observe that the two solutions have similar writing times (1.782 s vs 1.855 s for objects of 1 MB and  $L_{Fog} = 20$  ms) Reading times are also similar except when small objects are used (256KB). In this case, accessing the DHT becomes costly with a lot of objects

Figure 2 shows the access time of each object for a given client when a 10 clients read 100 objects of 256KB. The  $L_{Core}$  latency is set to 200 ms (potential latency for a mobile Fog site) and  $L_{Fog}$  latency is set to 10 ms. We observe that

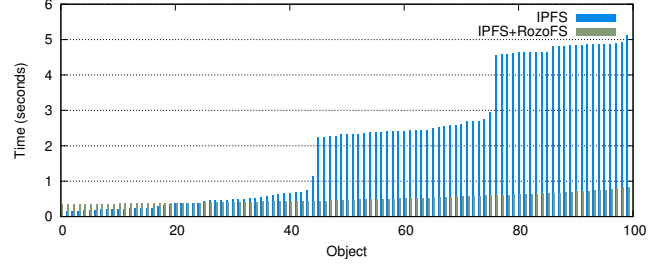


Fig. 2: Reading time of each object for a given client and a given iteration (100 × 256 KB).

access times are lower more stable when IPFS is coupled with RozoFS (approximately 0.5 s). On the contrary when IPFS is used alone, access times varies a lot depending on the number of hops needed to be done to reach the node storing the location of the object in the DHT.

### IV. CONCLUSION AND PERSPECTIVES

We showed coupling IPFS to a Scale-Out NAS improves the local reading times without increase of the writing time. The coupling also limits the amount of network traffic exchanged between the sites and enables IPFS to work in case of network partitioning.

A future work may be to do the same test by considering access of objects stored on a remote site. We also consider replacing the global DHT used to locate the objects by the Locator/ID Separation Protocol (LISP) and a Software Defined Networks (SDN) controller may be introduced to place each object on the most appropriated node.

### REFERENCES

- [1] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog Computing and Its Role in the Internet of Things," in *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, ser. MCC '12, 2012, pp. 13–16.
- [2] B. Confais, A. Lebre, and B. Parrein, "Performance Analysis of Object Store Systems in a Fog/Edge Computing Infrastructures," in *IEEE Cloud-Com*, Luxembourg, Dec. 2016.
- [3] J. Benet, "IPFS - Content Addressed, Versioned, P2P File System," Protocol Labs Inc., Tech. Rep., 2014.
- [4] D. Pertin, S. David, P. Évenou, B. Parrein, and N. Normand, "Distributed File System based on Erasure Coding for I/O Intensive Applications," in *4th International Conference on Cloud Computing and Service Science*, Barcelona, Spain, Apr. 2014.
- [5] K. Sui, M. Zhou, D. Liu, M. Ma, D. Pei, Y. Zhao, Z. Li, and T. Moscibroda, "Characterizing and Improving WiFi Latency in Large-Scale Operational Networks," in *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '16. New-York, USA: ACM, 2016, pp. 347–360.
- [6] J. Huang, F. Qian, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck, "A Close Examination of Performance and Power Characteristics of 4G LTE networks," in *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '12. New York, NY, USA: ACM, 2012, pp. 225–238.